The Church-rosser theorem in orthogonal combinatory reduction systems

Zurab Khasidashvili

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

1992

25ème anniversaire

N° 1825

Programme 2
Calcul Symbolique, Programmation
et Génie logiciel

THE CHURCH-ROSSER THEOREM IN ORTHOGONAL COMBINATORY REDUCTION SYSTEMS

Zurab KHASIDASHVILI

Décembre 1992

RR-1825

The Church-Rosser theorem in Orthogonal Combinatory Reduction Systems

Zurab Khasidashvili
INRIA-Rocquencourt, 78153, France

Abstract

We introduce a notion of *Combinatory Reduction Systems* (CRSs) by extending the notion of Term Rewriting Systems (TRSs) with variable-binding and substitution mechanisms, define *orthogonal* CRSs (OCRSs), and prove a strict version of the *Church-Rosser* theorem for all OCRSs. Our notion of OCRSs is almost equivalent to that of Klop [6], Kennaway and Sleep [4], and Van Raamsdonk [10]. Different proofs of the Church-Rosser theorem are in Klop [6] and in Van Raamsdonk [10].

Le Théorème de Church-Rosser dans les Systèmes de Réduction Combinatoire

Résumé

On introduit une notion de *Systèmes de Réduction Combinatoire* (SRCs) en étendant la notion de Système de Réécriture de Termes (SRTs) à l'aide d'un mécanisme de liaison de variable et de substitution. On définit les SRC *orthogonaux* (SRCOs) et on prouve une version stricte du théorème de *Church-Rosser*. Notre notion de SRCOs est équivalente à celle de Klop [6], Kennaway et Sleep [4] et Van Raamsdonk [10]. Differentes preuves du théorème de Church-Rosser sont dans Klop [6] et dans Van Raamsdonk [10].

1 Introduction

Combinatory Reduction Systems have been introduced to provide a uniform framework for reductions with substitutions, as in the λ-calculus and its extensions [2]. Several formalisms are proposed in Klop [6], Kennaway and Sleep [4], and Van Raamsdonk [10]. They are extensions of Term Rewriting Systems [3, 7] by means of variable binding and substitution mechanisms. Restricted notions of CRSs were first introduced in Pkhakadze [9] and Aczel [1]. Here we describe a different notion of CRSs, based on the syntax of [9]. It is introduced in Khasidashvili [5] under the name of Expression Reduction Systems. The expressive power of our CRSs is almost equivalent to that of other notions of CRSs, despite of the differences in term-construction, variable-binding, and substitution mechanisms. By removing some "depth" restrictions in Aczel's definition of contraction schemes [1], one gets another equivalent notion of CRSs.

We illustrate the differences between syntaxes of the above CRS formats by the following example from [9].

$$\int_e^s t\, dx$$

In the syntax of Aczel [1], $\int$ is a *function form* of *arity* $(0,0,1)$, which indicates that $\int$ takes three arguments and binds a variable in the third argument. The integral is written as

$$\int(e, s, (x)t).$$

In the syntax of Kennaway and Sleep [4] and Van Raamsdonk [10], $\int$ is a ternary *function symbol*. The integral is written as

$$\int(e, s, [x]t),$$

where $[x]$ stands for *abstraction* by x. Note that $[x]t$ is now a term by itself, contrary to Aczel's, where the binding variable is attached to the integral form. All free occurrences of x in the term t are bound in $[x]t$.

Klop [6] uses an applicative format. In his syntax, $\int$ is a *constant*. The integral is written as

$$(((\int e)s)([x]t)),$$

where ( ) expresses the *application operation* and [ ] stands for *abstraction*.

In the syntax of Pkhakadze [9] and our syntax, $\int$ is a *quantifier sign* with *arity* $(1,3)$ and *scope indicator* (3). The first component in the arity indicates the number of binding variables of a quantifier $\int x$, and the second component specifies the number of arguments of $\int x$. The scope indicator specifies the arguments of $\int x$ in which $\int x$ binds free occurrences of the variable x. In our notation, the integral is written as

$$\int x\ (s, e, t).$$

So $\int x$ binds all free occurrences of x in t.

Following [9], we use *operator signs of substitution* S^{n+1} $(n = 1, 2, \ldots)$ and *S-rules* to express substitution, instead of β-rules of the λ-calculus, used in [6], [4] and [10]. The S-rules are written a

$$S^{n+1}a_1 \ldots a_n A_1 \ldots A_n A \to (A_1/a_1, \ldots, A_n/a_n)A,$$

where each A_i is a *term metavariable*, which ranges over terms, and a_i is a *metavariable* for variables. Therefore our term metavariables do not have arity, and the representation of rewrite rules, originated from logic and the λ-calculus, are syntactically closer to the original rules. For example, the β-rule is written

$$\beta : Ap(\lambda aA, B) \to (B/a)A,$$

where Ap is a binary function for application, λ is a quantifier sign for abstraction with arity $(1,1)$ and scope indicator (1), and A and B are term-metavariables. In Klop [6] the β-rule is written

$$((\lambda([x]Z_1(x)))Z_0) \to Z_1(Z_0),$$

and in Kennaway and Sleep [4] and Van Raamsdonk [10] the β-rule is written

$$Ap(\lambda([x]Z_1(x)), Z_0) \to Z_1(Z_0),$$

where Z_1 is a unary metavariable and Z_0 is a 0-ary metavariable. For any n-ary metavariable Z^n, $Z^n(x_1, \ldots, x_n)$ denotes an arbitrary term that may contain free occurrences of variables $x_1, \ldots, x_n$, and $Z^n(s_1, \ldots, s_n)$ stands for the same term where the free occurrences of variables $x_1, \ldots, x_n$ are replaced by $s_1, \ldots, s_n$ respectively.

Klop proves in [6] that orthogonal CRSs (OCRSs) have the *Church-Rosser* (CR) property. That is, for any coinitial reductions $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$ from a term t to terms s and e respectively, a common reduct o of s and e exists. Moreover, he shows that there is a canonical way to construct reductions $P' : s \twoheadrightarrow o$ and $Q' : e \twoheadrightarrow o$. This property is called CR^+ in [6]. Klop's proof of CR^+ is very complicated. Van Raamsdonk [10] gives an elegant proof of CR, which is similar to the Tait and Martin-Löf proof of confluence for the λ-calculus.

The proof of CR theorem, presented here, is a generalization of our earlier proof of CR theorem for a subclass of OCRSs, sketched in [5]. Apart from existence of reductions $P' : s \twoheadrightarrow o$ and $Q' : e \twoheadrightarrow o$ for any coinitial reductions $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$, we show that $P + P'$ (the concatenation of P and P') and $Q + Q'$ are *strictly equivalent*. That is, descendants of all subterms of t under $P + P'$ and $Q + Q'$ are the same in o. This is particularly useful to study normalizing and perpetual (i.e., anti-normalizing) strategies. Our proof resembles Klop's proof of CR^+ and is more simple.

The method used in our proof is the following. It is easy to see that developments in an OCRS R can be represented as reductions in an OCRS $\underline{R}$ the rules of which are obtained from R-rules by underlining their head-symbols (Lemma 3.1). To prove termination of $\underline{R}$-reductions, we split $\underline{R}$ into $\underline{R}_{fS} = \underline{R}_f \cup \underline{S}$ with the TRS-part $\underline{R}_f$ and the substitution part $\underline{S}$ (Definition 2.9), and associate to each $\underline{R}$-reduction an $\underline{R}_{fS}$-reduction (Definitions 2.9 and 2.10). Then we define a non-erasing version $\underline{R}^{\mu}_{fS}$ of $\underline{R}_{fS}$ (Definition 3.3) and prove that each term in $\underline{R}^{\mu}_{fS}$ has exactly one normal form (Lemmas 3.2 and 3.3). This implies that $\underline{R}^{\mu}_{fS}$ is strongly normalizable (i.e., all reductions terminate). Since to each $\underline{R}_{fS}$-reduction one can associate an $\underline{R}^{\mu}_{fS}$-reduction of the same length (Lemmas 3.5 and 3.6), $\underline{R}_{fS}$ is also strongly normalizable. (This idea of reducing termination proofs on proofs of existence of a normal form was first used in Nederpelt [8] for a typed λ-calculus). As a corollary, we have the Finite Developments theorem (FD) for OCRSs (Theorem 3.2). To prove the strict version of CR^+, we define a notion of descendant for subterms (Definitions 2.6, 2.8 and 2.11) and the notion of strict equivalence for reductions (Definition 2.12). After checking that one step reductions in OCRSs "strictly commute" (Lemmas 4.1-4.5), we prove that all coinitial, complete developments are strictly equivalent (Theorem 4.1). This fact is used finally to prove the strict version of CR^+ (Theorem 4.2).

2 Orthogonal Combinatory Reduction Systems

Definition 2.1

(a) Let Σ be an alphabet, comprising *(object) variables* $v_0, v_1, \ldots$; *function symbols*, also called *simple operators*; and *operator signs* or *quantifier signs*. Each function symbol has an *arity* $k \in N$ (indicating the number of arguments it needs), and each operator sign σ has an *arity* (m, n) with $m, n \neq 0$ such that for any sequence $x_1, \ldots, x_m$ of pairwise distinct variables $\sigma x_1 \ldots x_m$ is a *compound operator* or a *quantifier* with *arity n*. Occurrences of $x_1, \ldots, x_m$ in $\sigma x_1 \ldots x_m$ are called *binding* variables. Each quantifier $\sigma x_1 \ldots x_m$, as well as corresponding quantifier sign σ and binding variables $x_1 \ldots x_m$, has a *scope indicator* $(k_1, \ldots, k_l)$ to specify the arguments in which $\sigma x_1 \ldots x_m$ binds all free occurrences of $x_1, \ldots, x_m$.

(b) A set $Ter(\Sigma)$ of *terms* over Σ is defined as follows:

1. $v_i \in Ter(\Sigma)$.
2. If δ is an n-ary (simple or compound) operator and $t_1, \ldots, t_n \in Ter(\Sigma)$, then $\delta t_1 \ldots t_n \in Ter(\Sigma)$.

Notation We use x, y, z for variables; t, s, e, o for terms, and σ, δ for operators and operator signs. We write $s \in t$ if s is a subterm of t.

Remark 2.1 In [5] we considered a more general alphabet, which may contain other kinds of symbols such as propositional variables, predicate symbols, and moreover, functional and predicate variables, and so on. In such alphabets we have two kinds of legitimate expressions — terms and formulas of types (sorts) *nat* and *bool* respectively. Simple operators apart from arity, compound operators apart from arity and scope indicator, have *logicality indicators* or *types*. For example, the type of a binary predicate is $nat * nat \rightarrow bool$, since it takes terms as arguments and gives a formula, and the type of Hilbert's operator τx is $bool \rightarrow nat$, since it operates on a formula A and gives a term $\tau x A$, which is supposed to be an object, satisfying $A(x)$. In this paper we restrict ourselves to the above simple alphabet to simplify exposition of definitions and proofs, which remain valid for richer alphabets. Sometimes we shall give examples in first order languages.

In our alphabet Σ, we gave to each quantifier sign σ a scope indicator, thus giving the same scope indicator to any quantifier $\sigma x_1 \ldots x_m$ with the head σ and to its binding variables $x_1, \ldots, x_m$. We could allow binding variables to have (in general) different scope indicators, e.g., allowing the binding variable x in $\sigma xyz\, s\, t\, e\, o$ (with σ of arity (3,4)) to bind free occurrences of x in s and e, the binding variable y to bind free occurrences of y in s and o, and the binding variable z to bind free occurrences of z in none of the terms s, t, e, o. We are not doing so for simplicity. We do not allow the operators to have a *variable arity*, i.e., capacity of taking any number of arguments, because such operators can be replaced by countable number of operators with fixed arity. Saying above that expressive power of our CRSs are almost equivalent to that of other CRSs, we meant that one can allow these kinds of symbols in the alphabet, if needed.

Definition 2.2

1. Let $M(\Sigma)$ be a countable set of *metavariables*, comprising metavariables $a, b, c, \ldots$ for variables, called *object metavariables*, and metavariables $A, B, C, \ldots$ for terms, called *term metavariables*. Object metavariables range over variables and term metavariables range over terms over Σ.

2. An *assignment* is a function $\theta : M(\Sigma) \to Ter(\Sigma)$ that maps each object metavariable to a variable, and each term metavariable to a term over Σ. We demand that an assignment maps different object metavariables to different variables, but may map different term-metavariables to the same term. If W is a word over $M(\Sigma)\bigcup\Sigma$ and θ is an assignment, then the result of replacement of metavariables by their values under θ is called *θ-instance* of W and is denoted by $W\theta$.

3. A *metaoperator* is a word over $M(\Sigma)\bigcup\Sigma$ each instance of which is an operator. The *arity* of a metaoperator, and the *scope indicator* of a metaquantifier and its binding variables and metavariables are defined analogously.

4. A set $MTer(\Sigma)$ of *metaterms* over $\Sigma\bigcup M(\Sigma)$ is defined as follows:

   (a) Object metavariables, term metavariables, and terms over Σ are *metaterms.*

   (b) If δ is an n-ary metaoperator and $t_1, \ldots, t_n \in MTer(\Sigma)$, then $\delta t_1 \ldots t_m \in MTer(\Sigma)$.

   (c) If $t_1, \ldots, t_n, t \in MTer(\Sigma)$ and $a_1, \ldots, a_n$ are pairwise different object metavariables, then $(t_1/a_1, \ldots, t_n/a_n)t$ is a *metaterm* and is called a *metasubstitution.* Subterms $t_1, \ldots, t_n$ are called *mobile arguments*, and $a_1, \ldots, a_n$ are called *binding variables* of the metasubstitution. The *scope* of binding metavariables $a_1, \ldots, a_n$ is t.

5. We use t, s, e, o for metaterms as well as for terms.

6. A metaterm not containing metasubstitutions is called a *simple metaterm.*

7. If t is a metaterm and θ is an assignment, then the *θ-instance $t\theta$* of t is the term obtained from t by replacing metavariables with their values under θ, and by further stepwise replacement (from right to left or inside out) of subterms of the form $(t_1/x_1, \ldots, t_n/x_n)t$ by the result of substitution of terms $t_1, \ldots, t_n$ for free occurrences of $x_1, \ldots, x_n$ in t (with renaming bound variables to avoid collision).

Example 2.1 Let $t = (A/a)((x + y)/z)(a + z)$ (i.e., $t = (A/a)(+xy/b) + ab)$. Obviously, t is a metaterm. Further, let θ be an assignment such that $\theta(a) = x$ and $\theta(A) = (5 * 3)$. Then $t\theta = ((5*3)+((5*3)+y)$, and it is obtained from $((5*3)/x)((x+y)/z)(x+z)$ by replacing $((x+y)/z)(x+z)$ with $(x + (x + y))$ and then by replacing $((5 * 3)/x)(x + (x + y))$ with $((5 * 3) + ((5 * 3) + y))$.

Definition 2.3

1. A *Combinatory Reduction System* is a pair (Σ, R), where Σ is an *alphabet*, described in Definition 2.1, and R is a (finite or infinite) set of *rewrite rules* $r : t \to s$, where t and s are metaterms such that

   (a) The metaterm t is not a metavariable.

   (b) Each term metavariable that occurs in s occurs also in t. The metaterm s may contain occurrences of object metavariables that do not occur in t. They are called *additional object metavariables.*

   (c) The metaterms t and s do not contain variables, and each occurrence of an object metavariable in t and s is bound.

(d) An occurrence of a term-metavariable A in s is in the scope of an occurrence of an object metavariable a in s iff an occurrence of A in t is in the scope of an occurrence of a in t.

2. Each rule $r : t \to s$ has an *admissible set of assignments* $AV(r)$ such that, for any assignment $\theta \in AV(r)$, occurrences of variables in $s\theta$ that correspond to additional object metavariables in s do not bind variables in subterms that correspond to term metavariables of s.

3. For any rule $r : t \to s$ in R and $\theta \in AV(r)$, $t\theta$ is a *redex* or an *r-redex* or an *R-redex*, and $s\theta$ is the *contractum* of $t\theta$.

Remark 2.2 Terms o and e are called *congruent* (written $o \approx e$) if o is obtained from e by renaming bound variables. The conditions in Definition 2.3 imply that, for any rule $r : t \to s$, if $\theta, \theta' \in AV(r)$, then $t\theta \approx t\theta'$ implies $s\theta \approx s\theta'$. Below we identify all congruent terms.

Notation We use u, v, w for redexes. If s is obtained from t by replacing a redex u by its contractum, then we say that s is obtained from t by *reducing* u or by *contracting* u and write $t \xrightarrow{u} s$. A sequence of the form $t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots$ is a *reduction*. We use P, Q for reductions. We write $P : t \twoheadrightarrow s$ if P denotes a reduction from t to s consisting of 0 or more steps. The *length*, i.e., the number of reduction steps, of P is denoted by $|P|$. If the last term in P coincides with the initial term in Q, then $P + Q$ denotes concatenation of reductions P and Q.

Example 2.2 A *Term Rewriting System* (TRS) is a pair (Σ, R), where Σ is an alphabet comprising (object) variables and function symbols (each having a fixed arity), and R is a set of rules $t \to s$, where t and s are terms such that

(a) The term t is not a variable.

(b) Each variable that occurs in s occurs also in t.

Obviously any TRS is a CRS that does not contain quantifier signs in its alphabet. Left-hand sides and right-hand sides of rules in TRSs are simple metaterms. The variables in rules of TRSs play the role of term metavariables in rules of CRSs.

Example 2.3 Operator signs $\exists$ and $\exists!$ for "there exists" and "there exists exactly one", having arity $(1, 1)$, can be defined using Hilbert's operator sign τ as follows:

$$\exists aA \to (\tau aA/a)A$$

$$\exists! aA \to \exists aA \wedge \forall a \forall b (A \wedge (b/a)A \Rightarrow a = b)$$

where $\forall$ is the quantifier sign with arity $(1, 1)$ and scope indicator (1) for "for any". Any assignment is admissible for the $\exists$-rule. An assignment θ is admissible for the $\exists!$-rule iff $b\theta \notin FV(A\theta)$. Obviously, b is an additional object metavariable.

Definition 2.4 Let $t \to s$ be a rule in a CRS R and θ be an assignment. Subterms of a redex $v = t\theta$ that correspond to term metavariables of t are the *arguments* of v, and the rest is the *pattern* of v. Subterms of v rooted at the pattern are called the *pattern-subterms* of v. If R is a simple CRS, then *arguments*, *pattern*, and *pattern-subterms* are defined analogously in the contractum $s\theta$ of v.

Definition 2.5 A rewrite rule $t \to s$ in a CRS R is *left-linear* if t is a simple metaterm and is *linear*, i.e., no term metavariable occurs more than once in t. R is *left-linear* if each rule in R is so. $R = \{r_i | i \in I\}$ is *non-ambiguous* or *non-overlapping* if in no term redex-patterns can overlap, i.e., if r_i-redex u contains an r_j-redex u' and $i \neq j$, then u' is in an argument of u, and the same holds if $i = j$ and u' is a proper subterm of u. R is *orthogonal* (OCRS) if it is left-linear and non-overlapping, and if v and w are any R-redexes such that w is in an argument of v and $v \xrightarrow{w} v'$, then v' is also a redex such that v and v' are instances of the same rule (i.e., with the same set of admissible assignments).

Definition 2.6 Let $t \xrightarrow{u} s$ in a simple OCRS and e be the contractum of u in s. For each argument t^* of u there are 0 or more arguments of e. We call them $(u-)$*descendants* of t^*. Correspondingly, subterms of t^* have 0 or more *descendants*. The *descendant* of each pattern-subterm of u that is not a variable is e. (We do not define descendants of "variable pattern-subterms", which are binding variables). It is clear what is to be meant under *descendants* of subterms that are not in u. The notion of *descendant* extends naturally to arbitrary reductions in simple OCRSs.

Definition 2.7 The CRS S is a CRS comprising rules of the form

$$S^{n+1}a_1 \dots a_n A_1 \dots A_n A \to (A_1/a_1, \dots, A_n/a_n)A, \;\; n = 1, 2, \dots,$$

where S^{n+1} is an operator sign with arity $(n, n+1)$, called *operator sign of substitution*, and $a_1, \dots, a_n$ and $A_1, \dots, A_n, A$ are pairwise distinct object and term metavariables respectively. Each assignment is admissible for any rule in S. We call $A_1, \dots, A_n$ the *mobile arguments* of S^{n+1}. (In the sequel we omit superscript in S^{n+1}. The arity of an operator sign S will be clear from the context).

Remark 2.3 Obviously, each step of S-reduction is a substitution of terms for free variables. In [5] we define an isomorphism between the class of all S-reductions in an appropriate alphabet (containing only binary S-operators) and the class of β-developments of λ-terms.

Definition 2.8 Let $u \xrightarrow{t} s$, where $u = Sx_1 \dots x_n t_1 \dots t_n t_0$, and let e be the contractum of u in s. For each mobile argument t_i of u $(i = 1, \dots, n)$ there are substituted occurrences of t_i in e. We call them *u-descendants* of t_i. By definition, they also are *u-descendants* of corresponding free occurrences of the variable x_i in t_0. Subterms in t_i have the same number (possibly none) of *descendants* in s. The *descendant* of u is e. It is clear, what is to be meant under *descendants* of subterms that are not in u, or are in t_0 and are not free occurrences of variables $x_1, \dots, x_n$. The notion of *descendant* extends naturally to S-reductions with 0 or more steps.

Definition 2.9 Let $R = \{r_i : t_i \to s_i | i \in I\}$ be an OCRS.

1. $R_f = \{r'_i : t_i \to s'_i | i \in I\}$, where s'_i is obtained from s_i by replacing all metasubstitutions of the form $(t_1/a_1, \dots, t_n/a_n)A$ with $\underline{S}^{n+1}a_1 \dots a_n t_1 \dots t_n A$ respectively. (We assume that symbols $\underline{S}^{n+1}$ do not occur in alphabet $\Sigma(R)$ of R and we add them to it to obtain the alphabet $\Sigma(R_f)$ of R_f. The *arity* and the *scope indicator* of $\underline{S}^{n+1}$ coincide with that of S^{n+1}. We shall omit the superscript in $\underline{S}^{n+1}$).

2. If R is simple, then $R_{fS} = R_f = R$. Otherwise $R_{fS} = R_f \cup \underline{S}$, where

$$\underline{S} = \{\underline{S}a_1 \dots a_n A_1 \dots A_n A \to (A_1/a_1, \dots, A_n/a_n)A \mid n = 1, 2, \dots\}.$$

3. $\underline{R} = \{\underline{r}_i : \underline{t}_i \to s_i | i \in I\}$, where $\underline{t}_i$ is obtained from t_i by underlining its head symbol. *Terms* in $\underline{R}$ are terms of R where (only) head-symbols of redexes may be underlined.

Remark 2.4 The functions $\_ : R \to \underline{R}$ and $f : R \to R_f$ commute. It is easy to see that $\underline{R}$ is closed under $\underline{R}$-reduction.

Definition 2.10 Let $r : t \to s$ be a rule in a CRS R and $r_f : t \to s'$ be its corresponding rule in R_f. For any step $e = C[t\theta] \xrightarrow{u} C[s\theta] = o$ in R (where θ is an admissible assignment for r) there is a reduction $P : e = C[t\theta] \to C[s'\theta] \twoheadrightarrow C[s\theta] = o$ in R_{fS}, where $C[s'\theta] \twoheadrightarrow C[s\theta]$ is the rightmost innermost normalizing $\underline{S}$-reduction. We call P the *expansion* of u and denote it by $Ex(u)$. The notion of *expansion* generalizes naturally to R-reductions with 0 or more steps.

Definition 2.11 Let $P : t \twoheadrightarrow s$ in an OCRS R and let $Q = Ex(P)$. It is clear from Definitions 2.6 and 2.8 what is to be meant under Q-descendants of subterms in t. We call a subterm $o' \in s$ a *P-descendant* of a subterm $o \in t$ if o' is a Q-descendant of o, and call o in this case the *P-ancestor* of o'.

Definition 2.12 We call the coinitial reductions $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow s$ *strictly equivalent* (written $P \approx Q$) if $s = e$ and P-descendants and Q-descendants of any subterm in t are the same in s and e.

Definition 2.13

1. We call an OCRS R *strictly Church-Rosser* or *strictly confluent* if for arbitrary coinitial reductions $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$ there are reductions $P' : s \twoheadrightarrow o$ and $Q' : e \twoheadrightarrow o$ such that $P + P' \approx Q + Q'$, i.e., the following diagram is *strictly commutative*.

$$\begin{tikzcd} t \arrow[r, two heads, "P"] \arrow[d, two heads, "Q"'] & s \arrow[d, two heads, "P'"] \\ e \arrow[r, two heads, "Q'"'] & o \end{tikzcd}$$

2. A term t in a CRS R is said to be in *normal form* (nf) or to be a *nf* if it does not contain redexes. If $s \twoheadrightarrow t$ and t is a nf, then t is called a *nf of s*. A term is called *weakly normalizable* if it has a nf, and is called *strongly normalizable* if it does not possess an infinite reduction. A CRS R is called *weakly normalizable* (resp. *strongly normalizable*) if each term in R is weakly normalizable (resp. strongly normalizable).

3 The Finite Developments Theorem

Definition 3.1 Let $t \xrightarrow{u} s$. Descendants of all redexes of t except u are also called *residuals*. By definition, u does not have *residuals* in s. The notion of *residual* of redexes extends naturally to reductions with 0 or more steps.

Notation We write $F \in t$, if F is a set of redexes in t. If $F \in t$ and $P : t \twoheadrightarrow s$, then F/P denotes the set of all residuals of redexes from F in s. If $F = \{u\}$, then we write u/P for $\{u\}/P$.

Definition 3.2 Let $F \in t_0$. We call $P : t_0 \rightarrow t_1 \rightarrow \ldots$ an *F-development* or a *development of F* if $u_i \in F_i = F/(u_0 + \ldots u_{i-1})$ for $i = 0, 1, \ldots$ If moreover u_i is the rightmost innermost redex from F_i (resp. $|P| < \infty$ and $F/P = \emptyset$), then we call P *canonical* (resp. *complete*) *F-development*. If F is the set of all redexes in t, then an F-development is called a *development of t*.

Notation If $F \in t$, then F will also denote the canonical complete F-development (it exists) and if $u \in t$, then u will also denote the reduction $t \xrightarrow{u} s$.

Lemma 3.1 Let t_0 be a term in an OCRS R and $F \in t_0$ be a set of redexes in t_0. Then for any F-development $P : t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots$ there is a reduction $Q : s_0 \xrightarrow{v_0} s_1 \xrightarrow{v_1} \ldots$ in $\underline{R}$ such that s_i is obtained from t_i by underlining head symbols of redexes from $F_i = F/(u_0 + \ldots u_{i-1})$, and u_i and v_i are corresponding redexes in t_i and s_i for all $i = 0, 1, \ldots$
Proof From Definitions 2.9 and 3.2.

Definition 3.3 Let $\underline{\Sigma}_{fS}$ be the alphabet of $\underline{R}_{fS}$ and $\underline{\Sigma}^{\mu}_{fS} = \underline{\Sigma}_{fS} \cup \{\mu^n \mid n = 0, 1, \ldots\}$, where μ^n is a fresh simple operator with arity n (we shall omit n in μ^n. The arity of a μ-operator will be clear from the context). *Terms* over $\underline{\Sigma}^{\mu}_{fS}$ are defined in the usual way with the restriction that underlined symbols may appear only as head-symbols of $\underline{R}^{\mu}_{fS}$-redexes. By definition, a step of $\underline{S}^{\mu}$*-reduction* is a replacement of an $\underline{S}$-redex (which is also an $\underline{S}^{\mu}$-redex) $u = \underline{S}x_1 \ldots x_n t_1 \ldots t_n t_0$ by $\mu t_{i_1} \ldots t_{i_k} t_0^*$, where $t_0^* = (t_1/x_1, \ldots, t_n/x_n)t_0$, i.e., t_0^* is the contractum of u in $\underline{S}$, and $t_{i_1}, \ldots, t_{i_k}$ are all arguments of u that do not have descendents in t_0^* (i.e., $x_{i_j} \notin FV(t_0)$ for $j = 1, \ldots, k$). The subterms $t_{i_1}, \ldots, t_{i_k}$ are called *erased arguments* of u or *u-erased arguments*, and $i_1, \ldots, i_k$ is called the *u-erased sequence*. A step of $\underline{R}^{\mu}_f$*-reduction* is a replacement of an $\underline{R}_f$-redex (which is also an $\underline{R}^{\mu}_f$-redex) $v = C[s_1, \ldots, s_m]$, where $C[\ ]$ is the context of v and $s_1, \ldots, s_m$ are its arguments, by $\mu s_{j_1} \ldots s_{j_l} s$, where s is the contractum of v in $\underline{R}_f$ and $s_{j_1}, \ldots, s_{j_l}$ are arguments of v that do not have descendants in s. The subterms $s_{j_1}, \ldots, s_{j_l}$ are called *v-erased arguments* and $j_1, \ldots, j_l$ is called the *v-erased sequence*. An $\underline{R}^{\mu}_{fS}$*-reduction* step of a term over $\underline{\Sigma}^{\mu}_{fS}$ is an $\underline{S}_{\mu}$-reduction step or an $\underline{R}^{\mu}_f$-reduction step. The notions of *descendant*, *residual*, *development*, etc. are defined analogously for $\underline{R}^{\mu}_{fS}$-reductions.

Notation $\underline{!}^{\mu}_f(t)$ denotes the result of the rightmost innermost normalizing $\underline{R}^{\mu}_f$-reduction of t; $\underline{!}^{\mu}_S(t)$ denotes the result of the rightmost innermost normalizing $\underline{S}^{\mu}$-reduction of t, and $\underline{!}^{\mu}_{fS}(t)$ denotes $\underline{!}^{\mu}_S(\underline{!}^{\mu}_f(t))$. Obviously, $\underline{!}^{\mu}_{fS}(t)$ is an $\underline{R}^{\mu}_{fS}$-nf of t. We call it the canonical $\underline{R}^{\mu}_{fS}$-nf of t.

Lemma 3.2 **(1)** All innermost normalizing $\underline{R}^{\mu}_f$-reductions of a term t end at the same term.

(2) All innermost normalizing $\underline{S}^{\mu}$-reductions of a term t end at the same term.
Proof (1) By induction on number n of redexes in t. If $n = 0$, then the lemma is obvious. Otherwise, let $P : t \xrightarrow{u} t_1 \rightarrow \ldots \rightarrow t_k$ and $Q : t \xrightarrow{v} s_1 \rightarrow \ldots \rightarrow s_l$ be any innermost normalizing $\underline{R}^{\mu}_f$-reductions and e be the common reduct of t_1 and s_1, i.e., $v/u : t_1 \twoheadrightarrow e$ and $u/v : s_1 \twoheadrightarrow e$. Then, by the induction assumption, $t_k = \underline{!}^{\mu}_f(t_1) = \underline{!}^{\mu}_f(e) = \underline{!}^{\mu}_f(s_1) = s_l$.
(2) can be proved analogously.

Lemma 3.3 Any term t over $\underline{\Sigma}^{\mu}_{fS}$ has exactly one $\underline{R}^{\mu}_{fS}$-nf.
Proof The proof is in a number of steps:

(a) Let s be a term over $\underline{\Sigma}^{\mu}_{fS}$, let v be an $\underline{R}^{\mu}_{f}$-redex in s, and u be an $\underline{S}^{\mu}$-redex in s. Then u and v commute, i.e., the following diagram is commutative,

$$\begin{tikzcd}
s \arrow[r, "u"] \arrow[d, "v"'] & e \arrow[d, twoheadrightarrow, "v/u"] \\
{} \arrow[r, twoheadrightarrow, "u/v"'] & {}
\end{tikzcd}$$

where u/v or v/u contain at most one step, depending on $v \in u$ or $u \in v$.

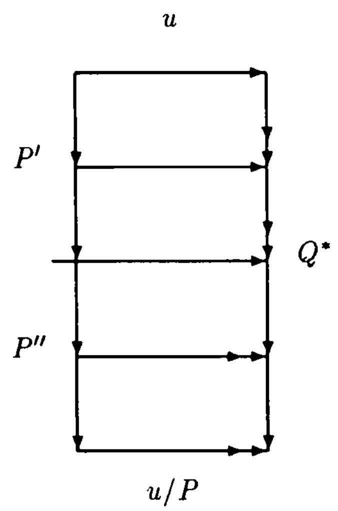
Proof of (a) is routine.

(b) Let s be a term over $\underline{\Sigma}^{\mu}_{fS}$ and u be an $\underline{S}^{\mu}$-redex in s. Then the following diagram is commutative,

$$\begin{tikzcd}
s \arrow[r, "u"] \arrow[d, twoheadrightarrow, "P"'] & e \arrow[d, twoheadrightarrow, "Q"] \\
\underline{!}^{\mu}_{f}(s) \arrow[r, twoheadrightarrow, "u/P"'] & \underline{!}^{\mu}_{f}(e)
\end{tikzcd}$$

where P and Q are innermost normalizing $\underline{R}^{\mu}_{f}$-reductions.

Proof of (b): By Lemma 3.2, we can assume that $P = P' + P''$, where in P' innermost $\underline{R}^{\mu}_{f}$-redexes are contracted first in mobile arguments of u (as long as possible) and then in the last argument of u, and in P'' only $\underline{R}^{\mu}_{f}$-redexes outside residuals of u are contracted. Then using (a) we can construct the following diagram



Obviously, Q^* is innermost.

(c) Let v be an innermost $\underline{R}^{\mu}_{f}$-redex in s and u be an arbitrary $\underline{R}^{\mu}_{f}$-redex in s. Then the following diagram is commutative,

$$\begin{tikzcd} s \arrow[r, "u"] \arrow[d, "v"'] & e \arrow[d, "v/u"] \\ {} \arrow[r, "u/v"'] & {} \end{tikzcd}$$

where u/v contains at most one element and v/u is innermost.

Proof of (c) is routine.

(d) Let u be an arbitrary $\underline{R}^{\mu}_{f}$-redex in s. Then the following diagram is commutative,

$$\begin{tikzcd} s \arrow[r, "u"] \arrow[d, "P"'] & e \arrow[d, "Q"] \\ \underline{!}^{\mu}_{f}(s) \arrow[r, "\emptyset"'] & \underline{!}^{\mu}_{f}(e) \end{tikzcd}$$

where P and Q are innermost normalizing $\underline{R}^{\mu}_{f}$-reductions.

(d) follows immediately from (c).

(e) Let u and v be $\underline{S}^{\mu}$-redexes in s, and v be in an argument of u. Then the following diagram is commutative,

$$\begin{tikzcd}
s \arrow[r, "u"] \arrow[d, "v"'] & e \arrow[d, "v/u"] \\
{} \arrow[r, "u'"'] & {}
\end{tikzcd}$$

where u' is the residual of u. Moreover, (α): if v is an innermost $\underline{S}^{\mu}$-redex in a mobile argument of u, whenever not all of mobile arguments of u are $\underline{S}^{\mu}$-nf's, and is an innermost $\underline{S}^{\mu}$-redex in the last argument of u otherwise, then v/u is innermost.

(e) can be checked similarly to the case of S-reductions. In addition, one has only to note that u and u' have the same erased sequences.

(f) Let u be an $\underline{S}^{\mu}$-redex in s. Then the following diagram is commutative,

$$\begin{tikzcd}
s \arrow[r, "u"] \arrow[d, two heads, "P"'] & e \arrow[d, two heads, "Q"] \\
!^{\mu}_{\underline{S}}(s) \arrow[r, two heads, "\emptyset"'] & !^{\mu}_{\underline{S}}(e)
\end{tikzcd}$$

where P and Q are innermost normalizing $\underline{S}^{\mu}$-reductions.

By (α), P can be chosen in such a way that Q is also innermost. Hence (f) follows from (e).

(g) Let $P' : s \twoheadrightarrow e$ be an $\underline{S}^{\mu}$-reduction. Then the following diagram is commutative,

$$\begin{tikzcd}
s \arrow[r, two heads, "P'"] \arrow[d, two heads, "P"'] & e \arrow[d, two heads, "Q"] \\
!^{\mu}_{\underline{S}}(s) \arrow[r, two heads, "\emptyset"'] & !^{\mu}_{\underline{S}}(e)
\end{tikzcd}$$

where P and Q are innermost normalizing $\underline{S}^{\mu}$-reductions.

(g) is a corollary of (f).

Now let $P^* : t = t_0 \to t_1 \to \ldots \to t_n$ be a normalizing $\underline{R}^{\mu}_{fS}$-reduction, $s_i = \underline{!}^{\mu}_{f}(t_i)$, and $e_i = \underline{!}^{\mu}_{S}(s_i)$. It follows from Lemma 3.2, from (b) and (d) that s_n is obtained from s_0 by an $\underline{S}^{\mu}$-reduction. Thus, by (g), $\underline{!}^{\mu}_{S}(s_0) = \underline{!}^{\mu}_{S}(s_n)$. Hence $t_n = s_n = e_n = e_0 = \underline{!}^{\mu}_{fS}(t)$, i.e., each $\underline{R}^{\mu}_{fS}$-nf of t coincides with the canonical one.

Notation $\|t\|_{\mu}$ denotes the number of occurrences of μ-symbols in t.

Lemma 3.4 $\underline{R}^{\mu}_{fS}$ is strongly normalizable.
Proof Let $P : t = t_0 \to t_1 \to \ldots$ be an $\underline{R}^{\mu}_{fS}$-reduction and $e = \underline{!}^{\mu}_{fS}(t)$. By Lemma 3.3, $e = \underline{!}^{\mu}_{fS}(t_i)$ for all $i = 1, 2, \ldots$ It is easy to see that $\|t_i\|_{\mu} \geq i$ and $\|t_i\|_{\mu} \leq \|e\|_{\mu}$. Thus $|P| \leq \|e\|_{\mu}$, i.e., P is finite.

Definition 3.4 Let $s = \mu t_1 \ldots t_n t_0 \in t$. Subterms $t_1, \ldots, t_n$, as well as subterms and symbols in $t_1, \ldots, t_n$, are called *μ-erased*, or more precisely *μ'-erased*, where μ' is the head symbol of s. The term obtained from t by removing all μ-erased symbols is denoted by $[t]_{\mu}$.

Lemma 3.5 Let R be an OCRS, let t be a term in $\underline{R}^{\mu}_{fS}$, $[t]_{\mu} = t'$ and $t' \xrightarrow{u'} s'$ in $\underline{R}_{fS}$. Then there is a redex $u \in t$ such that $t \xrightarrow{u} s$ in $\underline{R}^{\mu}_{fS}$ and $[s]_{\mu} = s'$.
Proof Let $e' \in s'$ be the contractum of u' and $e \in s$ be the contractum of u. It suffices to show that $[e]_{\mu} = e'$. We show this for the case when u' is an $\underline{S}$-redex. The case when u' is an $\underline{R}_f$-redex is similar, but easier. Let $u = \underline{S}x_1 \ldots x_n t_1 \ldots t_n t_0$. Then $e = \mu t_{i_1} \ldots t_{i_k} t_0^*$, where $t_0^* = (t_1/x_1, \ldots, t_n/x_n)t_0$ and $t_{i_1}, \ldots, t_{i_k}$ are erased arguments of u. So $[e]_{\mu} = [t_0^*]_{\mu}$. It is clear that $u' = [u]_{\mu} = \underline{S}x_1 \ldots x_n [t_1]_{\mu} \ldots [t_n]_{\mu}[t_0]_{\mu}$ and $e' = ([t_1]_{\mu}/x_1, \ldots, [t_n]_{\mu}/x_n)[t_0]_{\mu}$. An occurrence o in t_0^* is μ-erased iff it is in a substituted occurrence t_i^* of t_i and is μ-erased in t_i^*, or o is outside of substituted occurrences of $t_1, \ldots, t_n$ in t_0^* and there is an occurrence μ' of μ-operator outside of those substituted occurrences such that o is μ'-erased. In the first case the ancestor of o is μ-erased in t_i, and in the second case the ancestor of o is μ-erased in t_0. So $[e]_{\mu} = [t_0^*]_{\mu} = ([t_1]_{\mu}/x_1, \ldots, [t_n]_{\mu}/x_n)[t_0]_{\mu} = e'$, and this completes the proof.

Lemma 3.6 Let R be an OCRS and $P : t_0 \to t_1 \to \ldots$ be a reduction in $\underline{R}_{fS}$. Then there is a reduction $Q : s_0 = t_0 \to s_1 \to \ldots$ in $\underline{R}^{\mu}_{fS}$ such that $[s_i]_{\mu} = t_i$ for all $i = 0, 1, \ldots$
Proof Lemma is a corollary of Lemma 3.5.

Lemma 3.7 Let R be an OCRS. Then $\underline{R}_{fS}$ is strongly normalizable.
Proof From Lemmas 3.4 and 3.6.

Lemma 3.8 Let v and u be S-redexes in a term t. Then $u + v/u \approx v + u/v$.
Proof Routine.

Theorem 3.1 The CRS S is strongly normalizable and strictly Church-Rosser.
Proof By Lemma 3.7, $\underline{S}$ is strongly normalizable. Hence, S is strongly normalizable. Now it remains to use Lemma 3.8 and "strict form" of Newman's Lemma. (Newman's Lemma just says that any abstract rewriting system that is strongly normalizable and in which any two coinitial steps commute is CR).

Corollary 3.1 Let F be a set of S-redexes in a term t. Then all complete F-developments are strictly equivalent.

Theorem 3.2 Let R be an OCRS. Then all developments in R are finite.
Proof By Lemma 3.7, for any reduction $P : t_0 \to t_1 \to \ldots$ in $\underline{R}$, its expansion $E(P) : t_0 \twoheadrightarrow t_1 \twoheadrightarrow \ldots$ in $\underline{R}_{fS}$ is finite. So $\underline{R}$ is strongly normalizable and the theorem follows from Lemma 3.1.

4 The Strict Church-Rosser Theorem

Notation If u is a redex in an OCRS R, then $f(u)$ denotes the corresponding redex and reduction in R_f.

Lemma 4.1 Let t be a term in an OCRS R, let u be an S-redex in t, and v be an R-redex in an argument of u. Then $u + f(v)/u \approx f(v) + u'$, where u' is the unique residual of u.
Proof Routine.

Lemma 4.2 Let t be a term in an OCRS R, let u be an S-redex in t, and v be an R-redex in an argument of u. Then $u + v/u \approx v + u'$, where u' is the unique residual of u.
Proof Let F be the set of all $\underline{S}$-redexes created by the step $f(v)$, and let $u'' = u/f(v)$. Then $v + u' \approx f(v) + F + u' \approx$ (by Corollary 3.1)$\approx$

$$\begin{tikzcd}
\cdot \arrow[r, "u"] \arrow[d, "f(v)"'] & \cdot \arrow[d, "f(v)/u"] \\
\cdot \arrow[r, "u''"] \arrow[d, "F"'] & \cdot \arrow[d, "F/u''"] \\
\cdot \arrow[r, "u'"'] & \cdot
\end{tikzcd}$$

$\approx f(v) + u'' + F/u'' \approx$ (by Lemma 4.1) $\approx u + f(v)/u + F/u'' \approx$ (since $f(v) + u'' \approx u + f(v)/u$ implies that F/u'' coincides with the set of all $\underline{S}$-redexes, created by $f(v)/u) \approx u + v/u$.

Lemma 4.3 Let t be a term in an OCRS R, let u be an S-redex in t, and let F be a set of R-redexes in t such that u is not inside redexes from F. Further, let P be an F-development (in R) and u' be the unique P-residual of u. Then there exists an F/u-development Q such that $P + u' \approx u + Q$. Moreover, if P is complete, then so is Q.
Proof By induction on $n = |P|$. The case $n = 0$ is trivial, so let $P = P' + v$, and let u'' be the unique P'-residual of u. By the induction assumption, there exists an F/u-development Q' such that $P' + u'' \approx u + Q'$.

$$\begin{tikzcd}
\cdot \arrow[r, "u"] \arrow[d, "P'"'] & \cdot \arrow[d, "Q'"] \\
\cdot \arrow[r, "u''"] \arrow[d, "v"'] & \cdot \arrow[d, "v/u''"] \\
\cdot \arrow[r, "u'"'] & \cdot
\end{tikzcd}$$

It follows from Lemma 4.2 that $u'' + v/u'' \approx v + u'$. Thus $u + Q' + v/u'' \approx P' + u'' + v/u'' \approx P' + v + u' = P + u'$. Now it is clear that we can take $Q = Q' + v/u''$. (Indeed, $Q' + v/u''$ is an F/u-development and if P is a complete F-development, then $F/P = F/(P + u') = \emptyset = (F/u)/Q$, i.e., Q is also a complete F/u-development).

Lemma 4.4 Let t be a term in an OCRS R, let u and v be R-redexes in t, let v be in an argument of u, and u' be a v-residual of u. Then $f(u) + v/f(u) \approx v + f(u')$.
Proof Routine.

Lemma 4.5 Let t be a term in an OCRS R, and u and v be R-redexes in t. Then there are complete u/v- and v/u-developments P and Q such that $u + P \approx v + Q$.
Proof If $u \cap v = \emptyset$ or $u = v$, then Lemma is obvious. Suppose that v is in an argument of u and $f(u) + w_1 + \ldots + w_n$ is the expansion of u. Using Lemmas 4.4 and 4.3 we can construct the following diagram

$$\begin{tikzcd}
t \arrow[r, "f(u)"] \arrow[d, "v"'] & \cdot \arrow[r, "w_1"] \arrow[d] & \cdot \arrow[r] \arrow[d] & \cdot \arrow[r, "w_n"] \arrow[d] & \cdot \arrow[d, "v/u"] \\
\cdot \arrow[r, "f(u)/v"'] & \cdot \arrow[r, "w_1'"'] & \cdot \arrow[r] & \cdot \arrow[r, "w_n'"'] & \cdot
\end{tikzcd}$$

where w_i' is the unique residual of w_i (under a complete $v/(f(u) + w_1 + \ldots + w_{i-1})$-development). By Lemmas 4.4 and 4.3 the right side of the diagram is a complete v/u-development and the bottom is the expansion of the R-step u/v, which is strictly equivalent to u/v. Now Lemma is obvious.

Theorem 4.1 Let t be a term in an OCRS R and F be a set of R-redexes in t. Then all complete F-developments are strictly equivalent.
Proof By induction on the least upper bound $n(F)$ of lengths of complete F-developments. If $n(F)$=0, then the theorem is obvious. Otherwise, let $P : t \xrightarrow{u} t_1 \to \ldots \to t_k$ and $Q : t \xrightarrow{v} s_1 \to \ldots \to s_l$ be any complete F-developments. By Lemma 4.5 there are complete $v/u-$ and u/v-developments P'' and Q'' such that $u + P'' \approx v + Q''$. Now, if P' is a complete $F/(u + P'')$-development, then by the induction assumption $P \approx u + P'' + P' \approx v + Q'' + P' \approx Q$.

Definition 4.1 Let $Q : t \twoheadrightarrow s$ and $t \xrightarrow{u} e$. Then the *residual* Q/u *of* Q *by* u is defined by induction on $|Q|$ as follows. If $Q = \emptyset_t$, then $Q/u = \emptyset_e$. If $Q = Q' + v$, then $Q/u = Q'/u + v/(u/Q')$.

Definition 4.2 Let $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$. Then the *residual* P/Q *of* P *by* Q and the *residual* Q/P *of* Q *by* P are defined by induction on $|P|$ as follows.
(1) If $P = \emptyset_t$, then $P/Q = \emptyset_e$ and $Q/P = Q$.
(2) If $P = P' + u$, then $P/Q = P'/Q + u/(Q/P')$ and $Q/P = (Q/P')/u$.
We shall write $P \sqcup Q$ for $P + Q/P$.

Lemma 4.6 Let t be a term in an OCRS R, let $t \xrightarrow{v} o$, $F \in t$, and let $P : t \twoheadrightarrow e$ be a complete F-development. Then $v \sqcup P \approx P \sqcup v$.
Proof It can be shown by induction on $|P|$ that $v \sqcup P$ and $P \sqcup v$ are complete $F \cup \{v\}$-developments. Hence the lemma follows from Theorem 4.1.

Lemma 4.7 Let $t \xrightarrow{v} e$ and $P : t \twoheadrightarrow o$ in an OCRS R. Then $v \sqcup P \approx P \sqcup v$.
Proof By induction on $|P|$. If $|P| = 0$, then the lemma is obvious. So let $P = P' + u$.

$$\begin{tikzcd}
\cdot \arrow[r, two heads, "P'"] \arrow[d, "v"'] & \cdot \arrow[r, "u"] \arrow[d, "v/P'"'] & \cdot \arrow[d, "(v/P')/u = v/P"] \\
\cdot \arrow[r, two heads, "P'/v"'] & \cdot \arrow[r, two heads, "u/(v/P')"'] & \cdot
\end{tikzcd}$$

Then, $v \sqcup P$ = (by Definition 4.2)= $v + P/v = v + (P' + u)/v$ = (by Definition 4.2)= $v + P'/v + u/(v/P')$ $\approx$(by the induction assumption) $\approx P' + v/P' + u/(v/P')$ $\approx$ (by Lemma 4.6) $\approx P' + u + (v/P')/u$ = (by Definition 4.1) = $P + v/P$ = (by Definition 4.2) = $P \sqcup v$.

Theorem 4.2 (Strict form of Church-Rosser theorem). Let R be an OCRS, and P and Q be coinitial reductions in R. Then $P \sqcup Q \approx Q \sqcup P$.
Proof By induction on $|P|$. If $|P| = 0$, then the theorem is obvious. So let $P = P' + u$.

$$\begin{tikzcd}
\cdot \arrow[r, two heads, "P'"] \arrow[d, two heads, "Q"'] & \cdot \arrow[r, "u"] \arrow[d, two heads, "Q/P'"'] & \cdot \arrow[d, two heads, "(Q/P')/u = Q/P"] \\
\cdot \arrow[r, two heads, "P'/Q"'] & \cdot \arrow[r, two heads, "u/(Q/P')"'] & \cdot
\end{tikzcd}$$

Then, $Q \sqcup P$ = (by Definition 4.2) = $Q + P/Q = Q + (P' + u)/Q$ = (by Definition 4.2)= $Q + P'/Q + u/(Q/P')$ $\approx$(by the induction assumption) $\approx P' + Q/P' + u/(Q/P')$ $\approx$ (by Lemma 4.7) $\approx P' + u + (Q/P')/u$ = (by Definition 4.1) = $P + Q/P$ = (by Definition 4.2) = $P \sqcup Q$.

Acknowledgments I enjoyed discussions with J. W. Klop, J.-J. Lévy, G. Mints, Sh. Pkhakadze, Kh. Rukhaia and V. Sazonov. I also would like to thank D. Doligez, I. Jacobs, G. Gonthier, D. Sangiorgi, G. Tagviashvili and K. Urbaitis for their help in preparation of this paper.

References

[1] Aczel P. "*A general Church-Rosser theorem*". Preprint, University of Manchester, 1978.

[2] Barendregt H.P. "*The Lambda Calculus, its Syntax and Semantics*". North-Holland, 1984.

[3] Dershowitz N., Jouannaud J.-P. "*Rewrite Systems*". In: J.van Leeuwen ed. Handbook of Theoretical Computer Science, Chapter 6, vol.B, 1990, p. 243-320.

[4] Kennaway J.R, Sleep M.R. "*Neededness is hypernormalizing in regular combinatory reduction systems*". Preprint, School of Informatica, University of East Anglia, Norwich, 1989.

[5] Khasidashvili Z. "*Expression Reduction Systems*". Proceedings of I.Vekua Institute of Applied Mathematics, vol.36, Tbilisi 1990, p. 200-220.

[6] Klop J.W. "*Combinatory Reduction Systems*". Mathematical Centre Tracts 127. Centre for Mathematics and Computer Science, Amsterdam, 1980.

[7] Klop J.W. "*Term Rewriting Systems*". In: S.Abramsky, D.Gabby, and T.Maibaum eds. Handbook of Logic in Computer Science, vol. II, Oxford University Press, 1992.

[8] Nederpelt R.P. "*Strong normalization for a typed lambda-calculus with lambda structured types*". Ph.D. Thesis, Eindhoven, 1973.

[9] Pkhakadze Sh. "*Some problems of the Notation Theory (in Russian)*". Proceedings of I.Vekua Institute of Applied Mathematics, Tbilisi 1977.

[10] Van Raamsdonk F. "*A simple proof of confluence for weakly orthogonal combinatory reduction Systems*". Report CS-R9234, CWI Amsterdam, 1992.

ISSN 0249-6399